

Amendments to the Claims

Please cancel Claims 24-29. Please amend Claims 18, 23 and 35. The Claim Listing below will replace all prior versions of the claims in the application:

Claim Listing

1. (Original) A computer implemented method for profiling execution of a computer program having functions, the method comprising the steps of:
 - a) augmenting the program by performing the following augmenting steps for substantially all functions of the program:
 - i) selecting a function;
 - ii) adding function entry profiling code to the selected function;
 - iii) adding function exit profiling code to the selected function; and
 - b) executing the augmented program and collecting profiling data for substantially all executed functions of the program.
2. (Original) The method of claim 1, wherein the program augmenting step comprises performing said augmenting steps for all functions of said program, and wherein the executing step comprises collecting profiling data for all executed program functions.
3. (Original) The method of claim 1, wherein the step of adding function entry profiling code comprises adding function entry parallel stack simulation code for maintaining a parallel stack indicating call chains, and wherein the step of adding function exit profiling code comprises adding function exit parallel stack simulation code for maintaining the parallel stack.
4. (Original) The method of claim 1, wherein the step of adding function exit profiling code comprises adding code to determine a self+descendants time for a call to the selected function and to incorporate the self+descendants time into profiling information indexed by an identifier for the selected function.

5. (Original) The method of claim 4, wherein the step of adding function exit profiling code comprises adding code to incorporate the self+descendants time into profiling information indexed by a partial call chain for the call to the selected function.
6. (Original) The method of claim 5, wherein the partial call chain identifies the selected function.
7. (Original) The method of claim 6, wherein the partial call chain also identifies an immediate caller of the selected function.
8. (Original) The method of claim 1, wherein the step of adding function exit profiling code comprises adding code to determine timing statistics for a call to the selected function and to incorporate the timing statistics into profiling information indexed by an identifier for the selected function and by a partial call chain for the call to the selected function.
9. (Original) The method of claim 1, wherein the program has relocatable object files containing substantially all of the functions of the program, and wherein the step of augmenting the program comprises adding profiling code to substantially all relocatable object files of the program.
10. (Original) The method of claim 9, wherein said relocatable object files comprise instructions, and wherein the step of augmenting the program further comprises for each function the steps of:
 - parsing the function instructions into blocks; and
 - adding block profiling code to each block.
11. (Original) The method of claim 9, wherein said relocatable object files comprise instructions, and wherein the step of augmenting the program further comprises for each function the steps of:
 - parsing the function instructions into basic blocks, wherein a basic block consists

of a contiguous sequence of at least one instruction of which the first instruction is a program control transfer destination, the last instruction is a program control transfer instruction, and any instructions between the first and last instructions are neither program control transfer destinations nor program control transfer instructions; and

for each basic block, performing the steps of selecting a basic block;

analyzing the selected basic block to determine a fixed number of clock cycles for the selected basic block; and

adding basic block profiling code so as to be executed whenever the basic block is executed, said basic block profiling code operating to add the determined fixed number of clock cycles to a clock cycle accumulator for the function.

12. (Original) The method of claim 11, wherein the added basic block profiling code operates to add the determined fixed number of clock cycles to a clock cycle accumulator for the selected basic block of the function.
13. (Original) The method of claim 11, wherein the step of analyzing the selected basic block comprises determining whether the selected basic block contains a variably determinate length operating system call, and wherein the step of adding basic block profiling code comprises, if the selected basic block contains a variably determinate length operating system call, the steps of
 - adding simulation maintenance code to simulate constraints that determine execution time for the variably determinate length operating system call;
 - adding simulation evaluation code to evaluate the constraints and to determine a number of clock cycles required to execute the variably determinate length operating system call; and
 - adding simulation profiling code to add the determined number of clock cycles to a clock cycle accumulator for the function.

14. (Original) The method of claim 13, wherein the added simulation profiling code operates to add the determined fixed number of clock cycles to a clock cycle accumulator for the selected basic block of the function.
15. (Original) The method of claim 11, wherein the step of analyzing the selected basic block comprises determining whether the selected basic block contains an operating system (OS) call, and wherein the step of adding basic block profiling code comprises, for at least one selected basic block containing an OS call, the OS call timing profiling steps of
 - adding timing start code to determine a start time of day immediately before executing the OS call;
 - adding timing end code to determine an end time of day immediately after executing the OS call;
 - adding OS call timing calculation code to determine from the start time and end time a number of clock cycles required to execute the OS call; and
 - adding OS call timing recording code to add the determined number of clock cycles to a clock cycle accumulator for the function.
16. (Original) The method of claim 15, wherein said OS call timing profiling steps are performed for substantially only basic blocks containing indeterminate length OS calls.
17. (Original) The method of claim 15, wherein the added OS call timing recording code operates to add the determined fixed number of clock cycles to a clock cycle accumulator for the selected basic block of the function.
18. (Currently amended) A computer implemented method for profiling execution of a computer program having a function, the method comprising the steps of:
 - a) augmenting the program by adding function profiling code to the function, said function profiling code, for each particular call to the function when executed, serving to determine an execution time for the particular call to the function and to

incorporate said execution time into profiling data for the function, said function profiling code including at least function entry profiling code;

- b) executing the augmented program and collecting said profiling data for the function; and
- c) presenting said profiling data to a user.

19. (Original) The method of claim 18, wherein the function has a varying execution time over a plurality of calls when said program is executed, wherein said profiling data comprises a global minimum execution time for the function and a global maximum execution time for the function over all calls to the function, and wherein said step of augmenting the program by adding function profiling code comprises adding function profiling code to determine the global minimum execution time for the function and the global maximum execution time for the function.
20. (Original) The method of claim 19, wherein substantially all calls to the function can be identified by one of a set of partial call chains, and wherein said step of augmenting the program by adding function profiling code comprises adding function profiling code to determine profiling data indexed by each partial call chain in said set of partial call chains.
21. (Original) The method of claim 20, wherein each said partial call chain identifies a caller of said function, wherein said profiling data comprises for each said caller a caller-wise minimum execution time for the function and a caller-wise maximum execution time for the function over all calls to the function originating said caller, and wherein said step of augmenting the program by adding function profiling code comprises adding function profiling code to determine for each said caller the caller-wise minimum execution time for the function and the caller-wise maximum execution time for the function.
22. (Original) A computer implemented method for profiling execution of a computer program having functions, the method comprising the steps of:

a) augmenting the program by performing the following augmenting steps for substantially all functions of the program:

- i) selecting a function;
- ii) adding function profiling code to the function, the function profiling code including
 - function entry profiling code added to the beginning of the selected function, the function entry profiling code including function entry parallel stack simulation code for maintaining a parallel stack indicating call chains,
 - function exit profiling code added to the end of the selected function, the function exit profiling code including function exit parallel stack simulation code for maintaining the parallel stack, and
 - function call profiling code for recording dynamic call information obtained from the parallel stack;

- b) executing the augmented program and collecting profiling data for substantially all executed functions of the program;
- c) processing the recorded dynamic call information to create a dynamic call graph; and
- d) presenting the dynamic call graph to a user.

23. (Currently amended) A computer implemented method for profiling execution of a computer program having functions, the functions having instructions, the method comprising the steps of:

- a) augmenting the program by adding profiling code to the program, including at least function entry profiling code;
- b) executing the augmented program and collecting profiling data for functions of the program, said profiling data including self+descendants times for the functions;
- c) constructing a call graph for the program; and

- d) displaying to a user a representation of the call graph including arcs connecting nodes, wherein each arc links a parent function to a child function and has a width corresponding to the self+descendants time for the child function.

24-29. (Cancelled)

- 30. (Original) A computer system configured as a new machine by a computer program, comprising means for performing the steps as recited in claim 1.
- 31. (Original) A computer system configured as a new machine by a computer program, comprising means for performing the steps as recited in claim 11.
- 32. (Original) A computer system configured as a new machine by a computer program, comprising means for performing the steps as recited in claim 18.
- 33. (Original) A computer system configured as a new machine by a computer program, comprising means for performing the steps as recited in claim 22.
- 34. (Original) A computer system configured as a new machine by a computer program, comprising means for performing the steps as recited in claim 23.
- 35. (Currently amended) A computer system configured as a new machine by a computer program, comprising means for performing the steps ~~as recited in claim 26~~ of:
 - a) augmenting the program by adding function profiling code to the program;
 - b) executing the augmented program and collecting profiling data for functions of the program, said profiling data including self+descendants times for the functions;
 - c) constructing a call graph showing function calls for the program;
 - d) providing a pruning time filter value; and

e) displaying to a user a representation of the call graph including arcs connecting nodes, wherein each arc links a parent function to a child function, wherein a set of arcs is automatically pruned from said displayed call graph representation, wherein each of said set of said automatically pruned arcs connects to a child function having a self+descendants time less than said pruning time filter value.